

Otimização de Processos (COQ897)

Prof. Argimiro R. Secchi

Segunda Lista de Exercícios - 2020

José Rodrigues Torraca Neto

6) Apresente a modelagem matemática do problema de otimização formulado no exercício 3.

▼ **Identificação de Sistemas - System Identification (SI)**

Um controle de processo bem sucedido tem como pré-requisito uma boa identificação. A qualidade do modelo utilizado impacta profundamente o desempenho do controlador. Geralmente, os modelos de processos dinâmicos não estão diretamente disponíveis, ou suas representações derivadas de leis físicas podem ser complicadas demais para a implementação do controlador devido à não-linearidade ou à presença de muitos parâmetros desconhecidos.

Por essas razões, os modelos lineares multivariáveis são muito utilizados na prática, e a maneira mais comum de obtê-los é coletando dados de entradas e saídas do processo em janelas de tempo específicas e utilizar algoritmos adequados para obter os parâmetros do modelo de processo.

Em resumo, o processo que envolve projetar experimentos, coletar dados e construir modelos é comumente referido como **identificação de sistema (SI)**. Assim, o objetivo do SI é encontrar um modelo que seja uma descrição boa o suficiente dos aspectos do sistema relevantes para determinado propósito, como por exemplo, o controle.

Pode-se definir a identificação de sistemas como a área de modelagem matemática a partir de dados experimentais. Mais detalhadamente, o procedimento de identificação consiste em, a partir de dados amostrados, determinar o melhor modelo (representação do sistema) dentro de uma classe de modelos, com base em determinado critério. Assim, é possível destacar três componentes básicos envolvidos na identificação de sistemas: o conjunto de dados, a classe de modelos e o critério para estimação.

O conjunto de dados é normalmente obtido a partir de testes projetados especificamente para o fim de identificação, de forma que ele contenha toda a informação necessária. A definição da classe de modelos é uma etapa que normalmente utiliza informações prévias do sistema. Por fim, o critério utilizado para a escolha do melhor modelo consiste nos algoritmos de identificação propriamente ditos. Após a identificação, é preciso verificar se o modelo é bom o suficiente para a aplicação pretendida, o que é feito na etapa de validação.

Os algoritmos de SI podem ser classificados em duas categorias principais: **Métodos de Erro de Predição (PEM)** e **Métodos de Identificação de Subespaços - Subspace State Space Systems Identification (SID)**.

Os métodos PEM visam identificar modelos de função de transferência de entrada-saída, enquanto que os métodos SID foram desenvolvidos para obter **modelos de espaço de estados linearmente invariantes no tempo (LTI)** diretamente dos dados de entrada / saída.

Métodos de Identificação de Subespaços - Subspace State Space Systems Identification (SID)

Os métodos SID englobam uma série de algoritmos de identificação que se baseiam em conceitos da teoria de sistemas, geometria e álgebra linear numérica. Um grande número de aplicações dos métodos SID para sistemas simulados e reais foram apresentados na literatura, sendo que essa técnica já alcançou um grau significativo de maturidade e aceitação.

Entre as vantagens do SID, pode-se destacar a sua simplicidade e robustez numérica, além da capacidade de tratar sistemas multivariáveis sem nenhum esforço adicional. O modelo é identificado no formato de espaço de estados, o que é conveniente para o modelo de controle preditivo (**MPC**). O único parâmetro necessário, a ordem do modelo, é obtido durante o procedimento de identificação. Ao contrário do que geralmente ocorre no PEM, no SID não é necessário resolver problemas de otimização não-lineares.

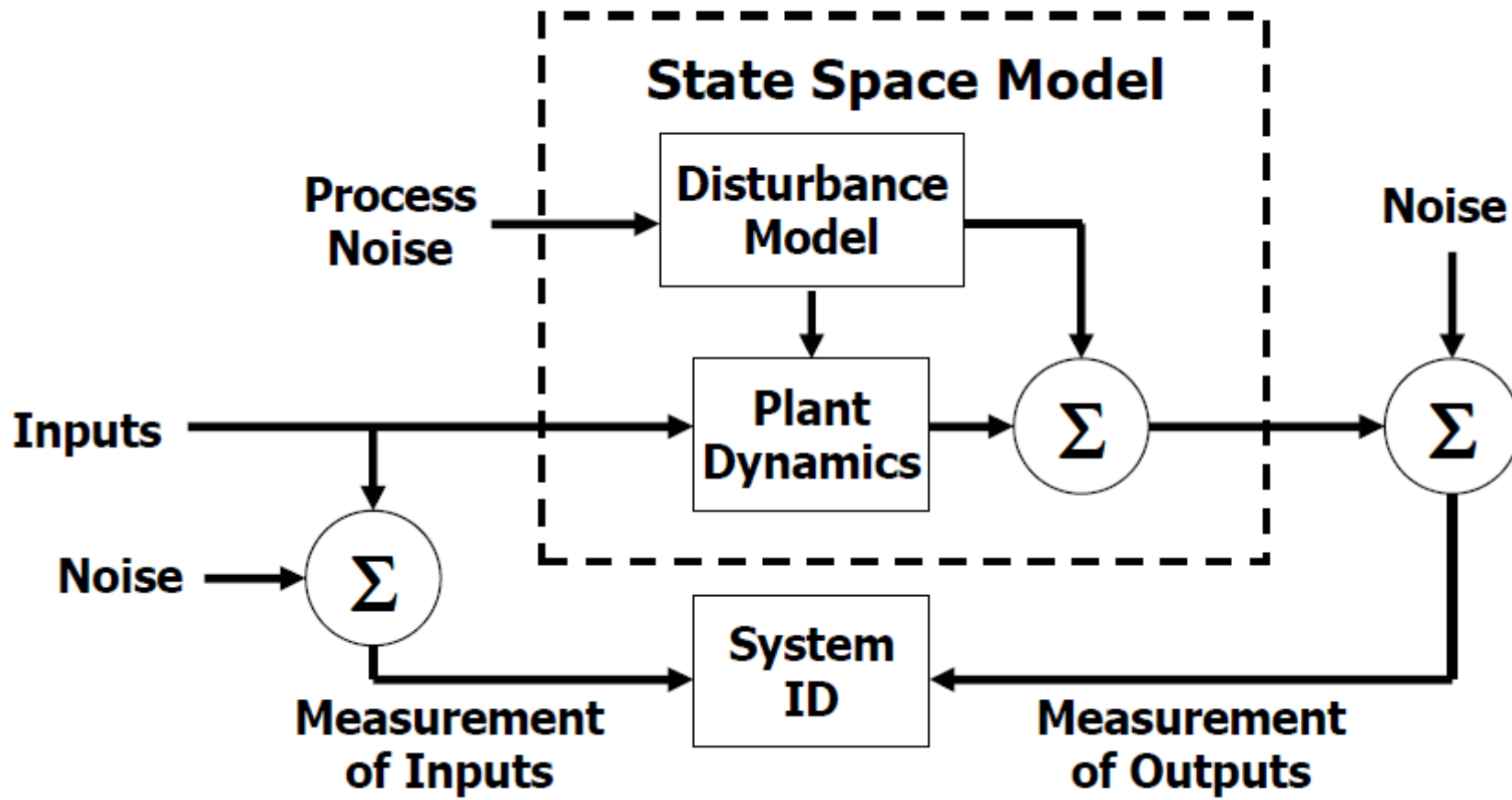
Em contra partida, os modelos obtidos via SID são sub-ótimos, no sentido que eles não são uma solução ótima de um problema de minimização do erro.

Além disso, os métodos SID são adequados para identificação de sistemas **MIMO** (Multiple Inputs Multiple Outputs). A complexidade da identificação para sistemas MIMO são iguais aos do SISO (Single Input Single Output). Não há necessidade de parametrização da estrutura MIMO, o que é uma propriedade muito atraente.

Logo, o nosso processo de SI será realizado utilizando um método SID. Os métodos SID não exigem que o usuário parametrize as matrizes do sistema antes de resolver um problema de otimização paramétrica e, como consequência, não sofrem de problemas relacionados a mínimos

locais que muitas vezes levam a resultados de identificação insatisfatórios.

A formulação de um problema SID consiste em, a partir de medições de entrada-saída, estimar um modelo espaço de estados de um processo MIMO, dado que pode haver perturbações de estado, perturbações de entrada e perturbações de saída. Esse processo pode ser melhor visualizado pela figura abaixo:



No SID, é considerado que o sistema com m entradas e l saídas é representado por um modelo em espaço de estados da seguinte forma:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + Du_k + v_k \end{aligned} \quad (1)$$

onde $x_k \in \mathbb{R}^n$, $w_k \in \mathbb{R}^{n \times x}$, $v_k \in \mathbb{R}^l$, são o estado do sistema, perturbações de estado e perturbações de saída, respectivamente. $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{l \times n}$, $D \in \mathbb{R}^{l \times m}$, são as matrizes do sistema.

Uma outra forma comum de representação de sistemas MIMO é pela sua forma de *inovação*:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + Ke(t) \\ y(t) &= Cx(t) + Du(t) + e(t) \end{aligned} \quad (2)$$

onde $K \in \mathbb{R}^{n \times l}$, é usualmente referido como o preditor de ganho de Kalman.

O problema a ser resolvido pelo SID pode ser descrito como: dado um conjunto grande de medições da entrada $u(t)$ e da saída $y(t)$, geradas pelo sistema desconhecido [2], determinar a ordem nx do sistema e estimar as matrizes do sistema (A ; B ; C ; D ; K) relativas a uma transformação de similaridade, a partir dos dados coletados.

As suas implementações podem ser classificadas em duas categorias:

- 1) Métodos tradicionais: **N4SID**, **MOESP** e **ORT**;
- 2) Métodos parcimoniosos: **PARSIM-P**, **PARSIM-S** e **PARSIM-K**.

Os métodos parcimoniosos removem termos não causais das projeções, impondo modelos causais. Os métodos mais antigos são PARSIM-S e PARSIM-P. No entanto, esses algoritmos são baseados na suposição de que a entrada é não correlacionada com a perturbação de saída, portanto, eles podem falhar para dados coletados em modo de malha fechada.

Os três métodos tradicionais, N4SID, MOESP e ORT, podem ser vistos como uma decomposição em valores singulares (SVD) de uma matriz ponderada. Por já terem sido testados extensivamente na literatura, além de usarem algoritmos que possibilitam aplicar identificação em malha fechada, esse grupo de métodos será considerado para o problema proposto.

O primeiro método, MOESP (multiple input multiple output output-error state space), foi derivado de um ponto de vista puramente estatístico, enquanto que o terceiro foi derivado da estrutura da teoria de realização estocástica. O segundo método reside entre os outros dois, pois usa elementos da teoria da realização, bem como da abordagem estatística (dados). No entanto, as implementações reais desses métodos funcionam com dados reais e, portanto, usam uma abordagem estatística. Para a identificação do sistema do problema proposto, apenas o método N4SID será considerado.

Retornando ao sistema em [2], a entrada u e a saída y são considerados como processos de segunda ordem, fracamente estacionários. Baseado nesta suposição, os espaços:

$$u = \overline{\text{span}} \{ \dots, u(t-1), u(t), u(t+1), \dots \}, \quad (3)$$

$$y = \overline{\text{span}} \{ \dots, y(t-1), y(t), y(t+1), \dots \} \quad (4)$$

podem ser definidos. A barra superior denota o fechamento com respeito ao produto interno definido por $E \{n\xi\}$, onde; $n, \xi \in \mathbf{u}$, e $E \{\cdot\}$ o valor esperado, ou esperança.

Além disso, permita que os subespaços

$$\mathbf{u}_t^- = \overline{span} \{ \dots, u(t-2), u(t-1) \}, \tag{5}$$

$$\mathbf{u}_t^+ = \overline{span} \{ u(t), u(t+1), \dots \} \tag{6}$$

sejam gerados pelos valores passados, bem como presentes e futuros de u .

Os espaços \mathbf{y}_t^+ e \mathbf{y}_t^- são igualmente descritos por y .

Seja a uma variável aleatória e $\mathbf{B} = \overline{span} \{b\}$ o intervalo da variável aleatória b , então a projeção ortogonal $\hat{E} \{a \mid \mathbf{B}\}$ é dada pelo valor esperado (esperança) condicional:

$$\hat{E} \{a \mid \mathbf{B}\} = E \{a \mid b\} = \mu_a + \sum_{ab} \sum_{bb}^{-1} (b - \mu_b) \tag{7}$$

Uma vez que ambos a e b são elementos de um espaço \mathbf{H} , há também uma projeção no complemento ortogonal de \mathbf{B} , calculada como:

$$\hat{E} \{a \mid \mathbf{B}^\perp\} = a - \hat{E} \{a \mid \mathbf{B}\} = a - \mu_a - \sum_{ab} \sum_{bb}^{-1} (b - \mu_b) \tag{8}$$

Agora, seja $\mathbf{C} = \overline{span} \{c\}$ o intervalo de uma terceira variável aleatória c cumprindo a condição $\mathbf{B} \cap \mathbf{C} = \{0\}$. A projeção oblíqua $\hat{E}_{\parallel \mathbf{C}} \{a \mid \mathbf{B}\}$ de a em \mathbf{B} ao longo de \mathbf{C} é dada pela matriz de covariância condicional:

$$\hat{E}_{\parallel \mathbf{C}} \{a \mid \mathbf{B}\} = E \left\{ \hat{E} \{a \mid \mathbf{C}^\perp\} \hat{E} \{\mathbf{B} \mid \mathbf{C}^\perp\} \right\} \times E \left\{ \hat{E} \{\mathbf{B} \mid \mathbf{C}^\perp\} \hat{E} \{\mathbf{B} \mid \mathbf{C}^\perp\} \right\}^{-1} b = \sum_{ab|c} \sum_{bb|c}^{-1} b \tag{9}$$

N4SID

A ideia principal do método **N4SID** é identificar o sistema dado pelas equações em [2], de modo que o subsistema conduzido pela entrada exógena u (referido como a parte "determinística"), bem como o sistema afetado pela perturbação e (referido como a parte "estocástica") irá compartilhar as matrizes A e C . A primeira etapa é o cálculo do espaço abrangido pelo estados. Assumindo liberdade de feedback e riqueza suficiente da entrada, este espaço é calculado através da projeção oblíqua (\vee denotando a soma vetorial):

$$\boldsymbol{\chi}_t^{+/-} = \hat{E}_{\parallel \mathbf{u}_t^+} \{ \mathbf{y}_t^+ \mid \mathbf{y}_t^- \vee \mathbf{u}_t^- \} \tag{10}$$

Uma vez que a implementação numérica produz não apenas um vetor base de $\boldsymbol{\chi}_t^{+/-}$, mas também a matriz de observabilidade estendida, os estados do sistema "determinístico" podem ser calculados por meios da matriz de observabilidade estendida e das saídas:

$$y_d(t) = \hat{E} \{ \mathbf{y}_t^+ \mid \mathbf{y}_t^- \vee \mathbf{u} \} \tag{10}$$

$$y_d(t+1) = \hat{E} \{ \mathbf{y}_{t+1}^+ \mid \mathbf{y}_{t+1}^- \vee \mathbf{u} \}$$

Esses estados são usados para resolver uma equação de mínimos quadrados do tipo:

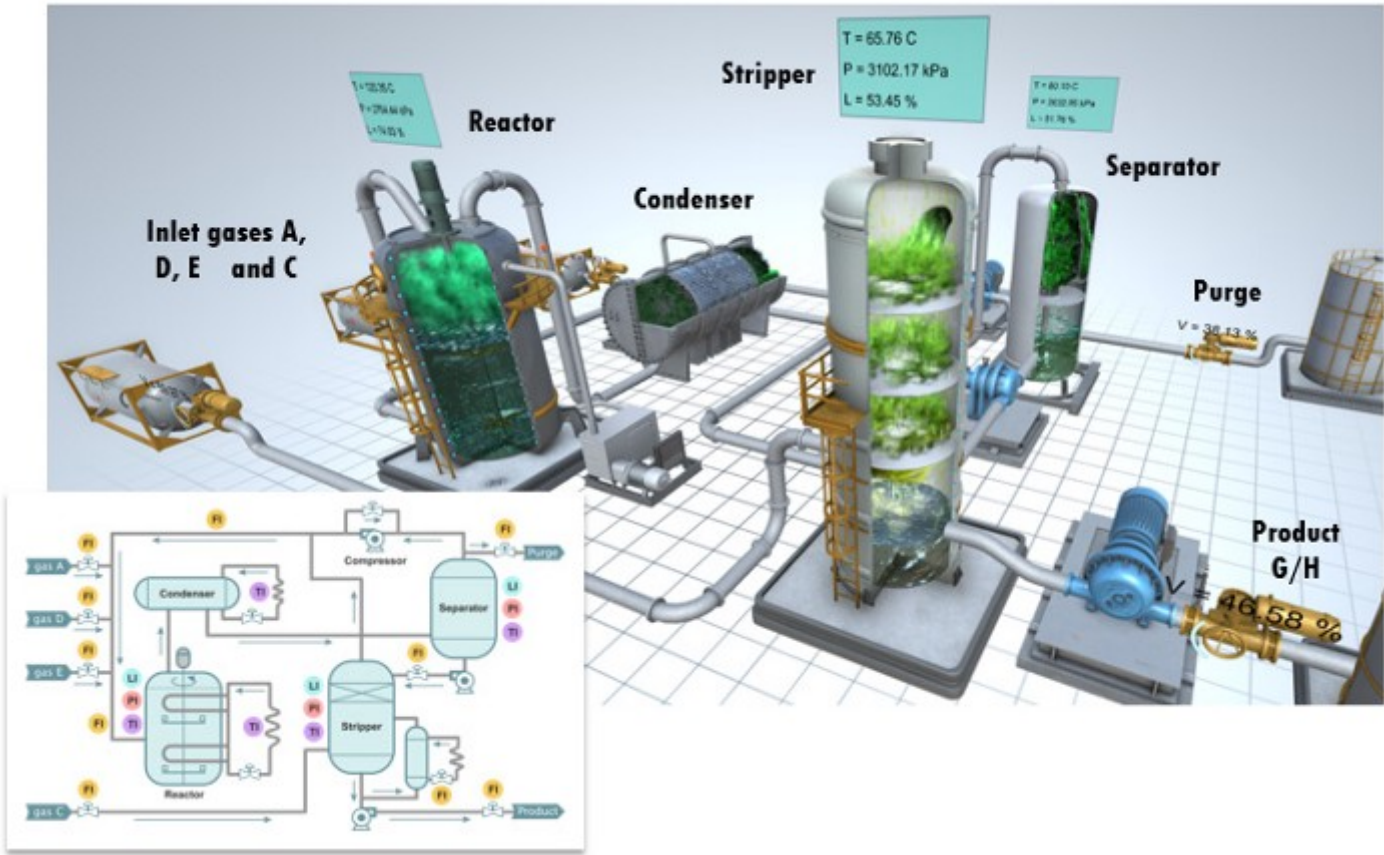
$$\min_{A,C,B,D} \left\| \begin{bmatrix} x^d(t+1) \\ y(t) \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x^d(t) \\ y(t) \end{bmatrix} \right\|_2 \tag{12}$$

para as matrizes do sistema. Os resíduos são usados para o cálculo do sistema "estocástico" e, portanto, para o cálculo de K .

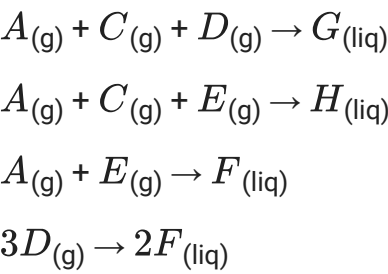
Método N4SID aplicado ao Tennessee Eastman Process (TEP)

▼ Tennessee Eastman Process

Como já dito anteriormente, o *Tennessee Eastman Process* é uma simulação realista de um processo químico da [Eastman Chemical Company](#) e foi introduzida por [DOWNS e VOGEL \(1993\)](#). É o benchmark mais utilizado na área de controle e monitoramento de processos, por fornecer um ambiente controlado para geração de dados, mas ainda assim realista.



Os reagentes gasosos A , C , D e E e o inerte B são alimentados ao reator, onde se formam os produtos G e H , no estado líquido, e o subproduto F , por meio de quatro reações simultâneas irreversíveis e exotérmicas:



A corrente de produto passa por um condensador e um separador líquido-vapor, respectivamente. O vapor resultante é reciclado por meio de um compressor, enquanto o condensado passa por um esgotador, da base do qual saem os produtos.

O processo apresenta 41 variáveis medidas e 12 variáveis manipuladas. Os tempos de amostragem variam entre 3, 6 e 15 minutos. São 20 perturbações na planta (falhas) pré-programadas na simulação, que variam entre aumento de variabilidade de algumas variáveis, agarramento de válvulas e tendência na cinética do processo, entre outras.

Segue uma descrição das variáveis e falhas (retirada do [link](#)):

Medições de processo contínuas

Variable	Description	unit
XMEAS(1)	A Feed (stream 1)	kscmh
XMEAS(2)	D Feed (stream 2)	kg/hr
XMEAS(3)	E Feed (stream 3)	kg/hr
XMEAS(4)	A and C Feed (stream 4)	kscmh
XMEAS(5)	Recycle Flow (stream 8)	kscmh
XMEAS(6)	Reactor Feed Rate (stream 6)	kscmh
XMEAS(7)	Reactor Pressure	kPa gauge
XMEAS(8)	Reactor Level	%
XMEAS(9)	Reactor Temperature	Deg C
XMEAS(10)	Purge Rate (stream 9)	kscmh
XMEAS(11)	Product Sep Temp	Deg C
XMEAS(12)	Product Sep Level	%
XMEAS(13)	Prod Sep Pressure	kPa gauge
XMEAS(14)	Prod Sep Underflow (stream 10)	m3/hr
XMEAS(15)	Stripper Level	%
XMEAS(16)	Stripper Pressure	kPa gauge
XMEAS(17)	Stripper Underflow (stream 11)	m3/hr
XMEAS(18)	Stripper Temperature	Deg C
XMEAS(19)	Stripper Steam Flow	kg/hr
XMEAS(20)	Compressor Work	kW
XMEAS(21)	Reactor Cooling Water Outlet Temp	Deg C
XMEAS(22)	Separator Cooling Water Outlet Temp	Deg C

Medições de processo amostradas

- Reactor Feed Analysis (Stream 6)

-
- Sampling Frequency = 0.1 hr
 - Dead Time = 0.1 hr
 - Mole %

Variable	Description
XMEAS(23)	Component A
XMEAS(24)	Component B
XMEAS(25)	Component C
XMEAS(26)	Component D
XMEAS(27)	Component E
XMEAS(28)	Component F

- Purge Gas Analysis (Stream 9)

-
- Sampling Frequency = 0.1 hr
 - Dead Time = 0.1 hr
 - Mole %

Variable	Description
XMEAS(29)	Component A
XMEAS(30)	Component B
XMEAS(31)	Component C
XMEAS(32)	Component D
XMEAS(33)	Component E
XMEAS(34)	Component F
XMEAS(35)	Component G
XMEAS(36)	Component H

- Product Analysis (Stream 11)

-
- Sampling Frequency = 0.25 hr
 - Dead Time = 0.25 hr
 - Mole %

Variable	Description
XMEAS(37)	Component D
XMEAS(38)	Component E
XMEAS(39)	Component F
XMEAS(40)	Component G
XMEAS(41)	Component H

Variáveis manipuladas

Variable	Description
XMV(1)	D Feed Flow (stream 2) (Corrected Order)
XMV(2)	E Feed Flow (stream 3) (Corrected Order)
XMV(3)	A Feed Flow (stream 1) (Corrected Order)
XMV(4)	A and C Feed Flow (stream 4)
XMV(5)	Compressor Recycle Valve
XMV(6)	Purge Valve (stream 9)
XMV(7)	Separator Pot Liquid Flow (stream 10)
XMV(8)	Stripper Liquid Product Flow (stream 11)
XMV(9)	Stripper Steam Valve
XMV(10)	Reactor Cooling Water Flow
XMV(11)	Condenser Cooling Water Flow
XMV(12)	Agitator Speed

Perturbações de processo

Variable	Description
IDV(1)	A/C Feed Ratio, B Composition Constant (Stream 4) Step

Variable	Description
IDV(2)	B Composition, A/C Ratio Constant (Stream 4) Step
IDV(3)	D Feed Temperature (Stream 2) Step
IDV(4)	Reactor Cooling Water Inlet Temperature Step
IDV(5)	Condenser Cooling Water Inlet Temperature Step
IDV(6)	A Feed Loss (Stream 1) Step
IDV(7)	C Header Pressure Loss - Reduced Availability (Stream 4) Step
IDV(8)	A, B, C Feed Composition (Stream 4) Random Variation
IDV(9)	D Feed Temperature (Stream 2) Random Variation
IDV(10)	C Feed Temperature (Stream 4) Random Variation
IDV(11)	Reactor Cooling Water Inlet Temperature Random Variation
IDV(12)	Condenser Cooling Water Inlet Temperature Random Variation
IDV(13)	Reaction Kinetics Slow Drift
IDV(14)	Reactor Cooling Water Valve Sticking
IDV(15)	Condenser Cooling Water Valve Sticking
IDV(16)	Unknown
IDV(17)	Unknown
IDV(18)	Unknown
IDV(19)	Unknown
IDV(20)	Unknown

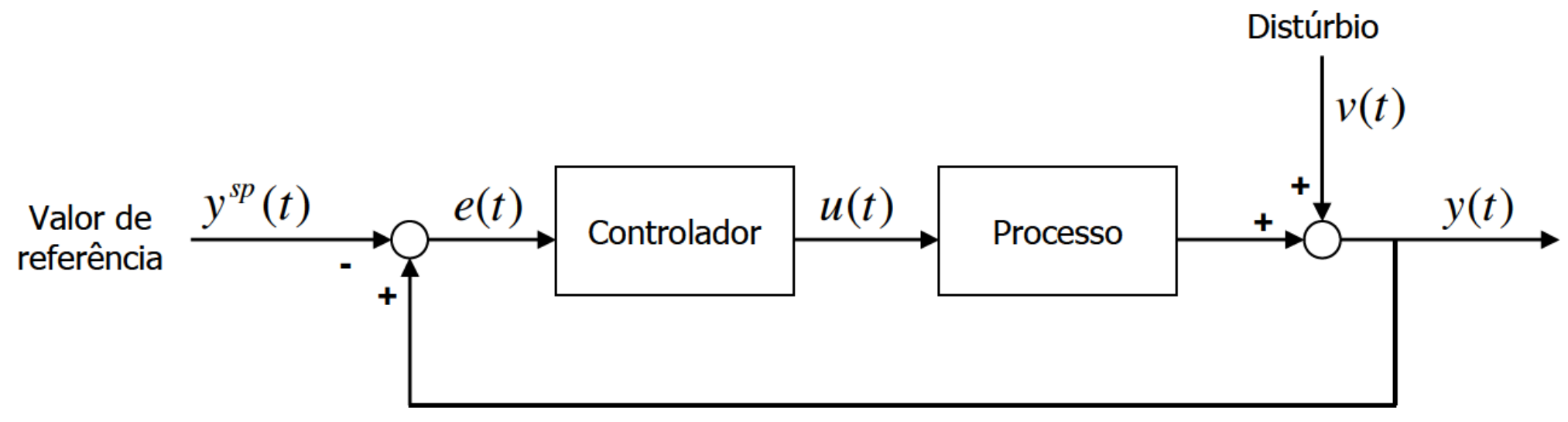
Sistema em Malha Fechada

É importante ressaltar que o processo opera sob **malha fechada**, ou seja, com a atuação de controladores automáticos. Nesse sentido, uma *falha* pode ser definida como uma perturbação no processo que a malha de controle não consegue gerenciar completamente.

Uma característica importante dessa técnica é a redução das perturbações devida à ação do controlador. Além disso, testes desse tipo são mais fáceis de executar, pois aspectos de especificação de produto e de segurança e estabilidade do processo são tratados automaticamente, eliminando atuações manuais. Ainda, segundo Hjalmarsson; Gevers e De Bruyne (1996), para o propósito de **controle**, modelos de qualidade superior são obtidos a partir de testes em malha fechada.

A desvantagem da identificação em malha fechada é que a entrada do processo é correlacionada com os distúrbios não medidos, sempre que a ação do controlador não é nula (FORSSELL; LJUNG, 1999). Problemas de identificabilidade podem surgir, porém eles são superados pela adição de excitação externa persistentemente excitante ou pelo uso de controladores com realimentação complexa, como o MPC, que possui características não-lineares e variantes no tempo.

Abaixo está a representação em blocos de um sistema em malha fechada:



▾ Simulações do sistema TEP

Boa parte dos estudos com a Tennessee Eastman utilizam um [pequeno conjunto de dados](#) já bem consolidado na literatura.

Os dados utilizados para este trabalho, no entanto, referem-se à uma extensão publicada por [RIETH et al. \(2017\)](#), que rodaram várias simulações e aumentaram consideravelmente o tamanho do conjunto, tornando-o mais adequado para aplicações *big data*.

Os dados encontram-se no formato RData, da [linguagem R](#). Para lê-los na linguagem Python, é necessário utilizar um módulo especial, como o [pyreadr](#):

```
%%time

# localização do dataset na sua máquina
PATH = '/kaggle/input/tennessee-eastman-process-simulation-dataset/'

train normal path = PATH+'TEP FaultFree Training RData'
```

```
train_normal_path = PATH+'TEP_FaultFree_Training.RData'
train_faulty_path = PATH+'TEP_Faulty_Training.RData'

test_normal_path = PATH+'TEP_FaultFree_Testing.RData'
test_faulty_path = PATH+'TEP_Faulty_Testing.RData'

train_normal_complete = pyreadr.read_r(train_normal_path)['fault_free_training']
#train_faulty_complete = pyreadr.read_r(train_fault_path)['faulty_training']

#test_normal_complete = pyreadr.read_r(test_normal_path)['fault_free_testing']
test_faulty_complete = pyreadr.read_r(test_faulty_path)['faulty_testing']

CPU times: user 1min 13s, sys: 6.72 s, total: 1min 20s
Wall time: 1min 20s
```

Inspeccionando os DataFrames, separados em conjuntos de treino e de teste (procedimento usual em técnicas de *Machine Learning*):

train_normal_complete

	faultNumber	simulationRun	sample	xmeas_1	xmeas_2	xmeas_3	xmeas_4	xmeas_5	xmeas_6	xmeas_7	...	xmv_2	xmv_3	...
0	0.0	1.0	1	0.25038	3674.0	4529.0	9.2320	26.889	42.402	2704.3	...	53.744	24.657	6
1	0.0	1.0	2	0.25109	3659.4	4556.6	9.4264	26.721	42.576	2705.0	...	53.414	24.588	5
2	0.0	1.0	3	0.25038	3660.3	4477.8	9.4426	26.875	42.070	2706.2	...	54.357	24.666	6
3	0.0	1.0	4	0.24977	3661.3	4512.1	9.4776	26.758	42.063	2707.2	...	53.946	24.725	5
4	0.0	1.0	5	0.29405	3679.0	4497.0	9.3381	26.889	42.650	2705.1	...	53.658	28.797	6
...
249995	0.0	500.0	496	0.29325	3640.1	4473.0	9.1949	26.867	42.379	2700.2	...	53.429	29.249	6
249996	0.0	500.0	497	0.29134	3625.7	4506.2	9.2109	26.889	42.291	2700.6	...	53.830	28.975	6
249997	0.0	500.0	498	0.29438	3600.2	4478.3	9.1957	26.820	42.448	2700.3	...	54.163	28.676	6
249998	0.0	500.0	499	0.25269	3683.5	4486.4	9.2832	27.188	42.757	2697.4	...	53.453	24.889	6
249999	0.0	500.0	500	0.25214	3648.2	4467.8	9.1344	26.886	42.534	2695.1	...	53.676	24.943	6

250000 rows × 55 columns

O DataFrame de treino foi gerado com 500 simulações, cada uma delas contendo 500 pontos de amostragem (totalizando 500*500 = 250000 pontos no conjunto de dados). As três primeiras colunas identificam os ID's da falha, da simulação e da amostra, respectivamente. As demais colunas contêm as medições de processo.

test_faulty_complete

	faultNumber	simulationRun	sample	xmeas_1	xmeas_2	xmeas_3	xmeas_4	xmeas_5	xmeas_6	xmeas_7	...	xmv_2	xmv_3	...
0	1	1.0	1	0.25171	3672.4	4466.3	9.5122	27.057	42.473	2705.6	...	54.494	24.527	...
1	1	1.0	2	0.25234	3642.2	4568.7	9.4145	26.999	42.586	2705.2	...	53.269	24.465	...
2	1	1.0	3	0.24840	3643.1	4507.5	9.2901	26.927	42.278	2703.5	...	54.000	24.860	...
3	1	1.0	4	0.25153	3628.3	4519.3	9.3347	26.999	42.330	2703.9	...	53.860	24.553	...
4	1	1.0	5	0.21763	3655.8	4571.0	9.3087	26.901	42.402	2707.7	...	53.307	21.775	...
...
9599995	20	500.0	956	0.26494	3719.6	4536.7	9.2265	26.875	42.401	2708.3	...	54.382	26.018	...
9599996	20	500.0	957	0.25252	3724.0	4494.3	9.1873	27.221	41.999	2706.0	...	54.236	25.098	...
9599997	20	500.0	958	0.25164	3700.8	4537.3	9.2514	26.659	42.180	2704.7	...	53.722	25.185	...
9599998	20	500.0	959	0.29097	3641.8	4525.2	9.3053	26.823	42.234	2705.0	...	54.185	28.771	...
9599999	20	500.0	960	0.29359	3622.2	4485.1	9.3070	26.922	42.610	2704.8	...	54.665	28.513	...

9600000 rows × 55 columns

O DataFrame de teste foi gerado com 500 simulações para cada um dos 20 cenários de falha, cada uma delas contendo 960 pontos de amostragem (totalizando 500*20*960 = 9600000 pontos no conjunto de dados). Como no conjunto de treino, as três primeiras colunas identificam os ID's da falha, da simulação e da amostra, respectivamente, e as demais colunas contêm as medições de processo.

Modelo de Controle Preditivo (MPC)

O modelo de controle preditivo (**MPC**) refere-se a uma classe de algoritmos computacionais de controle que utilizam um modelo explícito para prever o comportamento das saídas futuras da planta. Essa tecnologia é amplamente utilizada na indústria de processos químicos e geralmente é a técnica padrão usada em aplicações de controle avançado (GONZÁLEZ; ADAM; MARCHETTI, 2008).

Existem basicamente três enfoques utilizados em MPC. Cada enfoque adota uma estrutura de modelo. As possíveis estruturas são: resposta ao degrau ou resposta ao impulso (**FIR**, do inglês Finite Impulse Response); **função de transferência**; e modelo em espaço de estados **negrito** (WANG; YOUNG, 2006). Os primeiros algoritmos de controle preditivo baseavam-se em resposta ao degrau ou FIR. Por esse motivo essa é a técnica mais empregada nos controladores comerciais. Entretanto, no meio acadêmico, o enfoque mais utilizado atualmente, e que tem se mostrado mais eficiente, é o MPC baseado em **modelo em espaço de estados**, que já foi apresentado na primeira seção, e que será utilizado neste trabalho.

Modelo em espaço de estados

O estado de um sistema dinâmico é um conjunto de variáveis que, com o conhecimento delas em determinado instante de tempo, em conjunto com a informação das entradas futuras, determina completamente o comportamento futuro do sistema (ODLOAK, 2008). Essas variáveis são geralmente agrupadas em um vetor denominado vetor de estados. Esse vetor funciona como uma ponte de interligação entre o passado e o futuro. Um sistema linear invariante no tempo (LTI, do inglês Linear Time Invariant) pode ser representado por um modelo em espaço de estados com intervalos de tempo discretos da forma descrita pelo sistema [2].

Duas observações importantes podem ser feitas acerca desse modelo:

1. O modelo apresentado é incremental em relação às entradas. Essa representação foi escolhida, pois no MPC o uso das entradas na forma incremental é uma condição necessária para um controle livre de off-set.
2. Para processos típicos da indústria química, a matriz D é nula, ou seja, não há realimentação direta das entradas nas saídas.

A identificação em subespaços, que foi detalhada nas seções anteriores, é uma forma de se obter as matrizes (A ; B ; C ; D ; K) do modelo em espaço de estados diretamente de dados coletados da planta.

Quando um modelo em espaço de estados é utilizado para predição n passos à frente, como é o caso em aplicações de MPC, é necessário, em cada período de amostragem, atualizar o estado predito com o estado atual da planta. Seja o estado predito pelo modelo no instante $t + 1$:

$$\hat{x}(t + 1) = A\hat{x}(t) + B\Delta u(t) \quad (13)$$

Se o modelo usado for perfeito e as condições iniciais forem exatas, o estado predito será idêntico ao estado real da planta, não sendo necessário nenhum tipo de atualização. Porém, na prática, isso não acontece. Nos casos em que os componentes do vetor de estados são propriedades medidas, a atualização é realizada facilmente por uma simples substituição do estado predito pelo estado real disponível. Nos demais casos é preciso utilizar um estimador de estados. A reconstrução do estado é feita com base no erro de predição da saída ($y - C\hat{x} - D\Delta u$) da seguinte forma:

$$\hat{x}(t + 1 | t) = A\hat{x}(t | t - 1) + B\Delta u(t) + K[y(t) - C\hat{x}(t | t - 1) - D\Delta u(t)] \quad (14)$$

A notação $\hat{x}(j | s)$ é usada para indicar a predição do estado no instante j com base nas medidas disponíveis até o instante s . O parâmetro K é a matriz de ganhos que deve ser propriamente escolhida, sendo determinada pelo estimador de estados. O cálculo da matriz K foi definido no final da seção N4SID.

Metodologia do MPC

Neste trabalho será utilizado o MPC de horizonte finito com modelo em espaço de estados na forma incremental, representado pelo sistema [2]. Será considerado apenas o caso de sistemas sem ação direta da entrada na saída ($D = 0$). A cada período de amostragem o controlador resolve o seguinte problema de otimização:

$$\min_{\Delta u_t} J_t = \sum_{j=1}^{np} (y(t + j | t) - y^{sp})^T Q (y(t + j | t) - y^{sp}) + \sum_{j=0}^{nc-1} \Delta u(t + j | t)^T R \Delta u(t + j | t) \quad (14)$$

sujeito às seguintes restrições:

$$-\Delta u_{max} \leq \Delta u(t + j | t) \leq \Delta u_{max}, \quad j = 0, \dots, nc - 1 \quad (15)$$

$$u_{min} \leq u(t + j | t) \leq u_{max}, \quad j = 0, \dots, nc - 1 \quad (16)$$

, onde:

$Q \in \mathbb{R}^{l \times l}$: matriz de pesos das saídas;
 $R \in \mathbb{R}^{m \times m}$: matriz de supressão das entradas;
 $y^{sp} \in \mathbb{R}^l$: vetor de referência (set – point) das saídas;
 $u_{min} \in \mathbb{R}^m$: valores mínimos para as entradas;
 $u_{max} \in \mathbb{R}^m$: valores máximos para as entradas;
 $\Delta u_{max} \in \mathbb{R}^m$: variação máxima das entradas em cada instante;
 np : horizonte de predição;
 nc : horizonte de controle;
 J_t : função custo do controlador no instante t ;
 $\Delta u_t = [\Delta u(t | t)^T \quad \Delta u(t + 1 | t)^T \quad \dots \quad \Delta u(t + nc - 1 | t)^T]^T$

O problema acima pode ser escrito na forma de programação quadrática (QP). Para isso, deve-se colocar as predições das saídas futuras em um único vetor. O próximo passo consiste em determinar a relação desse vetor com as variáveis do problema (Δu_t) e com o modelo em espaço de estados.

A partir das equações de [2] do modelo e considerando $D = 0$, pode-se escrever:

$$\begin{aligned}
 y(t + 1 | t) &= Cx(t + 1 | t) = CAx(t) + CB\Delta u(t | t) \\
 y(t + 2 | t) &= CAx(t + 1 | t) + CB\Delta u(t + 1 | t) = CA^2x(t) + CAB\Delta u(t | t) + CB\Delta u(t + 1 | t) \\
 y(t + 3 | t) &= CA^3x(t) + CA^2B\Delta u(t | t) + CAB\Delta u(t + 1 | t) + CB\Delta u(t + 2 | t) \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 y(t + j | t) &= CA^jx(t) + CA^{j-1}B\Delta u(t | t) + CA^{j-2}B\Delta u(t + 1 | t) + \dots + CB\Delta u(t + j - 1 | t)
 \end{aligned}$$

No MPC de horizonte finito considera-se que a partir do instante $k + nc - 1$ as entradas permanecem constantes, ou seja, $\Delta u(t + nc | t) = \Delta u(t + nc + 1 | t) = \dots = 0$.

Então, pode-se escrever:

$$\begin{bmatrix} y(t + 1 | t) \\ y(t + 2 | t) \\ \vdots \\ y(t + nc | t) \\ y(t + nc + 1 | t) \\ \vdots \\ y(t + np | t) \end{bmatrix} = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{nc} \\ CA^{nc+1} \\ \vdots \\ CA^{np} \end{bmatrix} x(t) + \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{nc-1}B & CA^{nc-2}B & \dots & CB \\ CA^{nc}B & CA^{nc-1}B & \dots & CAB \\ \vdots & \vdots & \ddots & \vdots \\ CA^{np-1}B & CA^{np-2}B & \dots & CA^{np-nc}B \end{bmatrix} \begin{bmatrix} \Delta u(t | t) \\ \Delta u(t + 1 | t) \\ \vdots \\ \Delta u(t + nc - 1 | t) \end{bmatrix}$$

A equação acima, que representa o vetor de predições das saídas futuras, pode ser resumida na forma:

$$Y(t) = \Psi x(t) + \Theta \Delta u_t \quad (17)$$

Também é necessário representar o vetor com as entradas futuras (u_t), presente na restrição [16], em função do vetor das variações das entradas futuras (Δu_t). Por definição tem-se:

$$\begin{aligned}
 u(t | t) &= \Delta u(t | t) + u(t - 1) \\
 u(t + 1 | t) &= \Delta u(t + 1 | t) + \Delta u(t | t) + u(t - 1) \\
 &\vdots \\
 &\vdots \\
 u(t + nc - 1 | t) &= \Delta u(t + nc - 1 | t) + \dots + \Delta u(t | t) + u(t - 1)
 \end{aligned}$$

As equações acima podem ser reescritas na forma:

$$\begin{bmatrix} u(t | t) \\ u(t + 1 | t) \\ \vdots \\ u(t + nc - 1 | t) \end{bmatrix} = \begin{bmatrix} I_m \\ I_m \\ \vdots \\ I_m \end{bmatrix} u(t - 1) + \begin{bmatrix} I_m & 0 & \dots & 0 \\ I_m & I_m & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_m & I_m & \dots & I_m \end{bmatrix} \begin{bmatrix} \Delta u(t | t) \\ \Delta u(t + 1 | t) \\ \vdots \\ \Delta u(t + nc - 1 | t) \end{bmatrix}$$

A relação acima representa o vetor de entradas futuras, que pode ser então resumido como:

$$u_t = \bar{I}u(t - 1) + \bar{M}\Delta u_t \quad (18)$$

,sendo $u_t = [u(t | t)^T \quad u(t + 1 | t)^T \quad \dots \quad u(t + nc - 1 | t)^T]^T$

Utilizando as equações [17] e [18], o problema de otimização representado pela função objetivo [14] e pelas restrições [15] e [16] pode ser escrito como:

$$\min_{\Delta u_t} J_t = \Delta u_t^T H_1 \Delta u_t + 2c_{f1}^T \Delta u_t + c_1 \quad (19)$$

sujeito à restrição:

$$A_{ineq1} \Delta u_t \leq b_{ineq1} \quad (20)$$

, onde:

$$H_1 = \Theta^T \bar{Q} \Theta + \bar{R};$$

$$c_{f1}^T = [\Psi x(t) - Y_{sp}]^T \bar{Q} \Theta;$$

$$c_1 = [\Psi x(t) - Y_{sp}]^T \bar{Q} [\Psi x(t) - Y_{sp}];$$

$$A_{ineq1} = [(I_{m,nc})^T \quad (-I_{m,nc})^T \quad (\bar{M})^T \quad (-\bar{M})^T]^T;$$

$$b_{ineq1} = [(\bar{I} \Delta u_{max})^T \quad (-\bar{I} \Delta u_{max})^T \quad (\bar{I}(u_{max} - u(t-1)))^T \quad (\bar{I}(u(t-1) - u_{min}))^T]^T;$$

$$Y_{sp} = [(y^{sp})^T \quad (y^{sp})^T \quad \dots \quad (y^{sp})^T]^T;$$

$$\bar{Q} = \begin{bmatrix} Q & 0 & \dots & 0 \\ 0 & Q & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q \end{bmatrix};$$

$$\bar{R} = \begin{bmatrix} R & 0 & \dots & 0 \\ 0 & R & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R \end{bmatrix};$$

Após o cálculo das ações de controle futuras pela resolução do problema de otimização [14]-[16], a primeira ação de controle ($u(t)$) é adicionada ao sistema. As saídas do processo são então medidas e com essas novas informações, no próximo período de amostragem, todo o procedimento é repetido.